

---

Subject: Re: [PATCH 2/13] Small preparations for namespaces  
Posted by [Pavel Emelianov](#) on Fri, 25 May 2007 13:21:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Serge E. Hallyn wrote:

> Quoting Pavel Emelianov (xemul@openvz.org):

>> Serge E. Hallyn wrote:

>>> Quoting Pavel Emelianov (xemul@openvz.org):

>>>> This includes #ifdefs in get/put\_pid\_ns and rewriting

>>>> the child\_reaper() function to the more logical view.

>>>>

>>>> This doesn't fit logically into any other patch so

>>>> I decided to make it separate.

>>>>

>>>> Signed-off-by: Pavel Emelianov <xemul@openvz.org>

>>>>

>>>> ---

>>>>

>>>> diff --git a/include/linux/pid\_namespace.h b/include/linux/pid\_namespace.h

>>>> index 169c6c2..7af7191 100644

>>>> --- a/include/linux/pid\_namespace.h

>>>> +++ b/include/linux/pid\_namespace.h

>>>> @@ -26,7 +26,9 @@ extern struct pid\_namespace init\_pid\_ns;

>>>>

>>>> static inline void get\_pid\_ns(struct pid\_namespace \*ns)

>>>> {

>>>> #ifdef CONFIG\_PID\_NS

>>>> kref\_get(&ns->kref);

>>>> #endif

>>>> }

>>>>

>>>> extern struct pid\_namespace \*copy\_pid\_ns(int flags, struct pid\_namespace \*ns);

>>>> @@ -34,12 +36,15 @@ extern void free\_pid\_ns(struct kref \*kre

>>>>

>>>> static inline void put\_pid\_ns(struct pid\_namespace \*ns)

>>>> {

>>>> #ifdef CONFIG\_PID\_NS

>>>> kref\_put(&ns->kref, free\_pid\_ns);

>>>> #endif

>>>> }

>>>>

>>>> static inline struct task\_struct \*child\_reaper(struct task\_struct \*tsk)

>>>> {

>>>> - return init\_pid\_ns.child\_reaper;

>>>> + BUG\_ON(tsk != current);

>>>> + return tsk->nsproxy->pid\_ns->child\_reaper;

>>>> }

>>>>

>>>> #endif /\* \_LINUX\_PID\_NS\_H \*/  
>>> This can't be bisect-safe, right? You can't just use  
>>> tsk->nsproxy->pid\_ns, as you've pointed out yourself.  
>> I can :) See - I have a proving BUG\_ON() here.  
>  
> I didn't know BUG\_ON()'s actually warded off bugs :)

It does not, but it says to code reader that this call expects something special. In this case - tsk is expected to be current always. And it is.

> You've tested this with the infamous NFS testcase?

What testcase do you mean?

> I don't see \*why\* it would work for you, but if you claim it does, I  
> guess you'd know better than I :)

I don't get you here. I've checked that the task passed to child\_reaper is current always. This BUG\_ON prevents later code from passing arbitrary task to it.

> -serge  
>

---